(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 320 015 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
18.06.2003 Bulletin 2003/25

(51) Int Cl.[7]: **G06F 1/00**

(21) Application number: **02258535.0**

(22) Date of filing: **11.12.2002**

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR IE IT LI LU MC NL PT SE SI SK TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **12.12.2001 US 339634 P**
12.02.2002 US 75194
26.04.2002 US 132712

(71) Applicant: **Pervasive Security Systems Inc.**
Menlo Park, California 94025 (US)

(72) Inventors:
- **Zuili, Patrick**
  Palo Alto, CA 94306 (US)
- **Vainstein, Klimenty**
  Morgan Hill, CA 95037-9518 (US)

(74) Representative: **Ablett, Graham Keith et al**
Ablett & Stebbing,
Caparo House,
101-103 Baker Street
London W1U 6FQ (GB)

(54) System and method for providing manageability to security information for secured items

(57) The present invention relates to improved approaches for accessing secured digital assets (e.g., secured items). In general, digital assets that have been secured (secured digital assets) can only be accessed by authenticated users with appropriate access rights or privileges. Each secured digital asset is provided with a header portion and a data portion, where the header portion includes a pointer to separately stored security information. The separately stored security information is used to determine whether access to associated data portions of secured digital assets is permitted. These improved approaches can facilitate the sharing of security information by various secured digital assets and thus reduce the overall storage space for the secured digital assets. These improved approaches can also facilitate efficient management of security for digital assets.

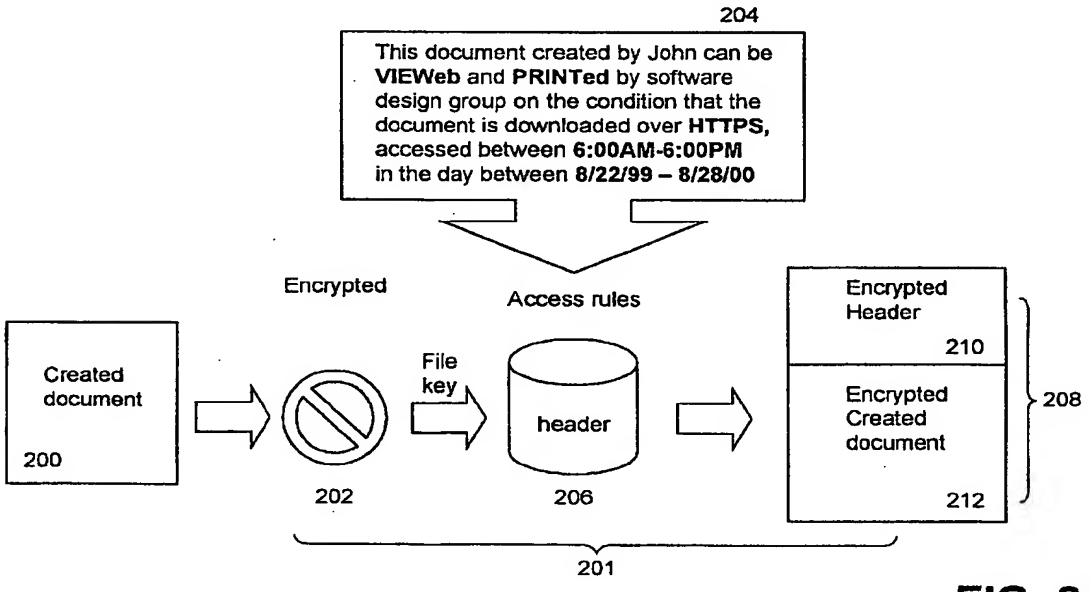

FIG. 2A

EP 1 320 015 A2

Description

[0001] The present invention relates to security systems for data and, more particularly, to security systems that protect data in an enterprise environment.

[0002] The Internet is the fastest growing telecommunications medium in history. This growth and the easy access it affords have significantly enhanced the opportunity to use advanced information technology for both the public and private sectors. It provides unprecedented opportunities for interaction and data sharing among businesses and individuals. However, the advantages provided by the Internet come with a significantly greater element of risk to the confidentiality and integrity of information. The Internet is a widely open, public and international network of interconnected computers and electronic devices. Without proper security means, an unauthorized person or machine may intercept any information travelling across the Internet and even get access to proprietary information stored in computers that interconnect to the Internet, but are otherwise generally inaccessible by the public.

[0003] There are many efforts in progress aimed at protecting proprietary information travelling across the Internet and controlling access to computers carrying the proprietary information. Cryptography allows people to carry over the confidence found in the physical world to the electronic world, thus allowing people to do business electronically without worries of deceit and deception. Every day hundreds of thousands of people interact electronically, whether it is through e-mail, e-commerce (business conducted over the Internet), ATM machines, or cellular phones. The perpetual increase of information transmitted electronically has lead to an increased reliance on cryptography.

[0004] One of the ongoing efforts in protecting the proprietary information travelling across the Internet is to use one or more cryptographic techniques to secure a private communication session between two communicating computers on the Internet. The cryptographic techniques provide a way to transmit information across an unsecure communication channel without disclosing the contents of the information to anyone eavesdropping on the communication channel. Using an encryption process in a cryptographic technique, one party can protect the contents of the data in transit from access by an unauthorized third party, yet the intended party can read the data using a corresponding decryption process.

[0005] A firewall is another security measure that protects the resources of a private network from users of other networks. However, it has been reported that many unauthorized accesses to proprietary information occur from the inside, as opposed to from the outside. An example of someone gaining unauthorized access from the inside is when restricted or proprietary information is accessed by someone within an organization who is not supposed to do so. Due to the open nature of the Internet, contractual information, customer data, executive communications, product specifications, and a host of other confidential and proprietary intellectual property remains available and vulnerable to improper access and usage by unauthorized users within or outside a supposedly protected perimeter.

[0006] Many businesses and organizations have been looking for effective ways to protect their proprietary information. Typically, businesses and organizations have deployed firewalls, Virtual Private Networks (VPNs), and Intrusion Detection Systems (IDS) to provide protection. Unfortunately, these various security means have been proven insufficient to reliably protect proprietary information residing on private networks. For example, depending on passwords to access sensitive documents from within often causes security breaches when the password of a few characters long is leaked or detected. Therefore, there is a need to provide more effective ways to secure and protect resources on private networks.

[0007] The invention relates to improved approaches for accessing secured digital assets (e.g., secured items). In general, digital assets that have been secured (secured digital assets) can only be accessed by authenticated users with appropriate access rights or privileges. Each secured digital asset is provided with a header portion and a data portion, where the header portion includes a pointer to separately stored security information. The separately stored security information is used to determine whether access to associated data portions of secured digital assets is permitted. These improved approaches can facilitate the sharing of security information by various secured digital assets and thus reduce the overall storage space for the secured digital assets. These improved approaches can also facilitate efficient management of security for the secured digital assets.

[0008] The invention can be implemented in numerous ways, including as a method, system, device, and computer readable medium. Several embodiments of the invention are discussed below.

[0009] As a method for accessing a secured file, one embodiment of the invention includes at least the acts of: obtaining the secured file to be accessed, the secured file having a header portion and a data portion; retrieving a security information pointer from the header portion of the secured file; obtaining security information for the secured file using the security information pointer; and permitting access to the secured file to the extent permitted by the security information.

[0010] As a computer readable medium including at least computer program code for accessing a secured item, one embodiment of the invention includes at least: computer program code for obtaining the secured item to be accessed, the secured item having a header portion and a data portion; computer program code for retrieving a security information pointer from the header portion of the secured item; computer program code for

obtaining security information for the secured item using the security information pointer; and computer program code for permitting access to the secured item to the extent permitted by the security information.

[0011] As a system for accessing a secured item, where the secured item has a header portion and an encrypted data portion, and where the header portion includes at least a pointer and an encrypted key, one embodiment of the invention includes at least: a storage device that stores security information for a plurality of different secured items, the pointer serving to locate the security information associated with secured item; a first decryption module that receives the encrypted key from the header portion of the secured item and decrypts the encrypted key to obtain a key; an access analyzer that determines whether the encrypted data portion is permitted to be accessed by a requestor based on the security information; and a second decryption module that decrypts the encrypted data portion using the key to produce an unencrypted data portion that the requestor is able to access, provided the access analyzer determines that the encrypted data portion is permitted to be accessed by a requestor.

[0012] As a data structure for a secured file, one embodiment of the invention includes at least a header portion and a data portion. The header portion contains at least a pointer to separately stored security information and a key. At least the key portion of the header portion is encrypted. The data portion contains at least encrypted data of the secured file.

[0013] Other objects, features, and advantages of the present invention will become apparent upon examining the following detailed description of an embodiment thereof, taken in conjunction with the attached drawings.

[0014] These and other features, aspects, and advantages of the present invention will become better understood with regard to the following description, appended claims, and accompanying drawings wherein:

FIG. 1A shows a basic system configuration in which the invention may be practised in accordance with an embodiment thereof.

FIG. 1B shows internal construction blocks of a computing device in which the invention may be implemented and executed.

FIG. 2A is a block diagram of securing a created document.

FIG. 2B is a block diagram of a secured item access system according to one embodiment of the invention.

FIG. 2C is a diagram of a representative data structure for a secured file.

FIG. 3 is a flow diagram of secured document access processing according to one embodiment of the invention.

FIG. 4A illustrates a data organization item according to one embodiment of the invention.

FIG. 4B illustrates exemplary tables for use with the data organization illustrated in FIG. 4A.

FIG. 5 is a block diagram of a file security management system according to one embodiment of the invention.

FIGs. 6A and 6B are flow diagrams of secured file portability processing according to one embodiment of the invention.

[0015] The present invention relates to improved approaches for accessing secured digital assets (e.g., secured items). In general, digital assets that have been secured (secured digital assets) can only be accessed by authenticated users with appropriate access rights or privileges. Each secured digital asset is provided with a header portion and a data portion, where the header portion includes a pointer to separately stored security information. The separately stored security information is used to determine whether access to associated data portions of secured digital assets is permitted. These improved approaches can facilitate the sharing of security information by various secured digital assets and thus reduce the overall storage space for the secured digital assets. These improved approaches can also facilitate efficient management of security for the secured digital assets.

[0016] Digital assets may include, but not be limited to, various types of documents, multimedia files, data, executable code, images and text. In the context of the present invention, digital assets may also include directories/folders as well as any OS-addressable resources (e.g. a thread to a port, or a device). The present invention is particularly suitable in an inter/intra enterprise environment.

[0017] In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will become obvious to those skilled in the art that the present invention may be practised without these specific details. The description and representation herein are the common means used by those experienced or skilled in the art to most effectively convey the substance of their work to others skilled in the art. In other instances, well-known methods, procedures, components, and circuitry have not been described in detail to avoid unnecessarily obscuring aspects of the present invention.

[0018] Reference herein to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment can be included in at least one embodiment of the invention. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment, nor are separate or alternative embodiments mutually exclusive of other embodiments. Further, the order of blocks in process flowcharts or diagrams representing one or more embodiments of the invention do not inherently indicate any particular order nor imply any limitations in the invention.

[0019] Embodiments of the present invention are discussed herein with reference to FIGs. 1A - 6B. However, those skilled in the art will readily appreciate that the detailed description given herein with respect to these figures is for explanatory purposes as the invention extends beyond these limited embodiments.

[0020] FIG. 1A shows a basic system configuration in which the present invention may be practised in accordance with one embodiment thereof. Documents or files may be created using an authoring tool executed on a client computer 100, which may be a desktop computing device, a laptop computer, or a mobile computing device. Exemplary authoring tools may include application programs such as Microsoft Office (e.g., Microsoft Word, Microsoft PowerPoint, and Microsoft Excel), Adobe FrameMaker and Adobe Photoshop.

[0021] According to one embodiment, the client computer 100 is loaded with a client module that is a linked and compiled, or interpreted, version of one embodiment of the present invention and is capable of communicating with a server 104 or 106 over a data network (e.g., the Internet or a local area network). According to another embodiment, the client computer 100 is coupled to the server 104 through a private link. As will be further explained below, a document or file created by an authoring tool can be secured by the client module. The client module, when executed, is configured to ensure that a secured document is secured at all times in a store (e.g., a hard disk or other data repository). The secured documents can only be accessed by users with proper access privileges. In general, an access privilege or access privileges for a user may include, but not be limited to, a viewing permit, a copying permit, a printing permit, an editing permit, a transferring permit, an uploading/downloading permit, and a location permit.

[0022] According to one embodiment, a created document is caused to go through an encryption process that is preferably transparent to a user. In other words, the created document is encrypted or decrypted under the authoring application so that the user is not aware of the process. A key (referred to herein as a user key) can be used to retrieve a file key to decrypt an encrypted document. Typically, the user key is associated with an access privilege for the user or a group of users. For a given secured document, only a user with a proper access privilege can access the secured document.

[0023] In one setting, a secured document may be uploaded via the network 110 from the computer 100 to a computing or storage device 102 that may serve as a central repository. Although not necessary, the network 110 can provide a private link between the computer 100 and the computing or storage device 102. Such link may be provided by an internal network in an enterprise or a secured communication protocol (e.g., VPN and HTTPS) over a public network (e.g., the Internet). Alternatively, such link may be simply provided by a TCP/IP link. As such, secured documents on the computer 100 may be remotely accessed.

[0024] In another setting, the computer 100 and the computing or storage device 102 are inseparable, in which case the computing or storage device 102 may be a local store to retain secured documents or receive secured network resources (e.g., dynamic Web contents, results of a database query, or a live multimedia feed). Regardless of where the secured documents or secured sources are actually located, a user, with proper access privilege, can access the secured documents or sources from the computer 100 or the computing or storage device 102 using an application (e.g., Internet Explorer, Microsoft Word or Acrobat Reader).

[0025] The server 104, also referred to as a local server, is a computing device coupled between a network 108 and the network 110. According to one embodiment, the server 104 executes a local version of a server module. The local version is a localized server module configured to service a group of designated users or client computers, or a location. Another server 106, also referred to as a central server, is a computing device coupled to the network 108. The server 106 executes the server module and provides centralized access control (AC) management for an entire organization or business. Accordingly, respective local modules in local servers, in coordination with the central server, form a distributed mechanism to provide distributed AC management. Such distributed access control management ensures the dependability, reliability and scalability of centralized AC management undertaken by the central server for an entire enterprise or a business location.

[0026] FIG. 1B shows internal construction blocks of a computing device 118 in which one embodiment of the present invention may be implemented and executed. The computing device 118 may correspond to a client device (e.g., computer 100, computing or storage device 102 in FIG. 1A) or a server device (e.g., server 104, 106 in FIG. 1A). As shown in FIG. 1B, the computing device 118 includes a central processing unit (CPU) 122 interfaced to a data bus 120 and a device interface 124. CPU 122 executes instructions to process data and perhaps manage all devices and interfaces coupled to data bus 120 for synchronized operations. The instructions being executed can, for example, pertain to drivers, operating system, utilities or applications. A device interface 124 may be coupled to an external device, such as the computing device 102 of FIG. 1A; hence, the secured documents therefrom can be received into memory 132 or storage 136 through data bus 120. Also interfaced to data bus 120 is a display interface 126, a network interface 128, a printer interface 130 and a floppy disk drive interface 138. Generally, a client module, a local module or a server module of an executable version of one embodiment of the present invention can be stored to storage 136 through floppy disk drive interface 138, network interface 128, device interface 124 or other interfaces coupled to data bus 120. Execution of such module by CPU 122 can cause the computing device 118 to perform as desired in the present invention. In

one embodiment, the device interface 124 provides an interface for communicating with a capturing device 125 (e.g., a fingerprint sensor, a smart card reader or a voice recorder) to facilitate the authentication of a user of the computing device 118.

[0027] Main memory 132, such as random access memory (RAM), is also interfaced to data bus 120 to provide CPU 122 with instructions and access to memory storage 136 for data and other instructions. In particular, when executing stored application program instructions, such as for document securing or document accessing, CPU 122 is caused to manipulate the data to achieve results contemplated by the program instructions. Read-Only Memory (ROM) 134 is provided for storing executable instructions, such as a basic input/output operation system (BIOS) for operation of keyboard 140, display 126 and pointing device 142 that may be present.

[0028] In one embodiment, the computing or storage device 102 is capable of storing secured items (e.g., secured files) in the main memory 132 or the storage 136. The main memory 132 provides non-persistent (i.e., volatile) storage for the secured items and the storage 136 provides persistent (i.e., nonvolatile) storage for the secured items. Hence, the computing or storage device 102, or more particularly, the main memory 132 and/or the storage 136, can act as a storage device for the secured items.

[0029] Referring now to FIG. 2A, a block diagram of securing a created document 200 is shown according to one embodiment of the invention. After the document 200 is created with an application or authoring tool and upon an activation of a "Save," "Save As" or "Close" command or automatic saving invoked by an operating system, the application itself or another application, the created document 200 is caused to undergo a securing process 201. The securing process 201 starts with an encryption process 202, namely, the document 200 that has been created or is being written into a store is encrypted by a cipher with a file key. In other words, the encrypted document could not be opened without the file key (i.e., a cipher key).

[0030] A set of access rules 204 for the document 200 is received and associated with a header 206. In general, the access rules 204 determine or regulate who and/or how the document 200, once secured, can be accessed. In some cases, the access rules 204 also determine or regulate when or where the document 200 can be accessed. Typically, a header is a file structure, small in size and includes, or perhaps links to, security information about a resultant secured document. Depending on an exact implementation, the security information can be entirely included in a header or pointed to by a pointer that is included in the header. According to one embodiment, the access rules 204, as part of the security information, are included in the header 206. According to another embodiment, the access rules 204, as part of the security information, are separately stored from the document 200 but referenced by one or more pointers or links therein. According to still another embodiment, the pointers in the header 206 can point to different versions of security information providing different access control depending on user's access privilege. The security information or the header 206 further includes a file key. Some or all of the header 206 can then be encrypted by a cipher with a user key associated with an authorized user to an encrypted header 210. The encrypted header 210 is attached to the encrypted document 212 to generate a secured document 208.

[0031] It is understood that a cipher may be implemented based on one of many encryption/decryption schemes. Examples of such schemes may include, but not be limited to, Data Encryption Standard algorithm (DES), Blowfish block cipher and Twofish cipher. Therefore, the operations of the present invention are not limited to a choice of those commonly-used encryption/decryption schemes. Any encryption/decryption scheme that is effective and reliable may be used. Hence, the details of encryption/decryption schemes are not further discussed herein so as to avoid obscuring aspects of the present invention.

[0032] To access the secured document 208, one needs to obtain the file key that is used to encrypt the document. To obtain the file key, one needs to be authenticated to get a user or group key and pass an access test in which the access rules in the security information are measured against the user's access privilege.

[0033] It should be noted that the header in a secured document may be configured differently than noted above without departing from the principles of the present invention. For example, a secured document may include a header with a plurality of encrypted headers, each can be accessible only by one designated user or a group users. Alternatively, a header in a secured document may include more than one set of security information or pointers thereto, each set being for one designated user or a group of users while a single file key can be used by all. Some or all of the access rules may be viewed or updated by users who can access the secured document.

[0034] In general, the encryption process and its counter process, decryption, are implemented in a filter or a software module that is activated when a secured document or item is involved. According to one embodiment in an operating system, the software module can be configured to control access to some digital assets (e.g., a port or a device) that may not be encrypted. However, an access to a secured port or device can trigger the software module to operate to control access thereto.

[0035] As will be further described below, to access a secured document, a user needs a user key or keys to decrypt the encrypted security information or at least a portion of the header first. In one embodiment, the key or keys are associated with a user's login to a local serv-

er or a central server. Appropriate access privileges associated with the user are validated if the user has been authenticated or previously registered with the server and properly logged in. Depending on the permission or the access privileges, the access rules for the secured document determine whether the contents of the document shall be revealed to the user.

[0036] According to one embodiment, the access rules are present in a markup language, such as HTML, SGML and XML. In a preferred embodiment, the markup language is Extensible Access Control Markup Language (XACML) that is essentially an XML specification for expressing policies for information access. In general, XACML can address fine-grained control of authorized activities, the effect of characteristics of the access requestor, the protocol over which the request is made, authorization based on classes of activities, and content introspection (i.e., authorization based on both the requestor and attribute values within the target where the values of the attributes may not be known to the policy writer). In addition, XACML can suggest a policy authorization model to guide implementers of the authorization mechanism.

[0037] In general, a document is encrypted with a cipher (e.g., a symmetric or asymmetric encryption scheme). Encryption is the transformation of data into a form that is impossible to read without appropriate knowledge (e.g., a key). Its purpose is to ensure privacy by keeping information hidden from anyone to whom it is not intended, even those who have access to other encrypted data. Decryption is the reverse of encryption. Encryption and decryption generally require the use of some secret information, referred to as a key. For some encryption mechanisms, the same key is used for both encryption and decryption; for other mechanisms, the keys used for encryption and decryption are different.

[0038] For the purpose of controlling the access to the document, the key or keys, referred collectively to as a file key, may be the same or different keys for encryption and decryption and are preferably included in the security information contained in or pointed to by the header and, once obtained, can be used to decrypt the encrypted document. To ensure that the key is not to be retrieved or accessible by anyone, the key itself is guarded by the access rules. If a user requesting the document has the proper access privileges that can be granted by the access rules, the key will be retrieved to proceed with the decryption of the encrypted document.

[0039] To ensure that the security information or the header is not readily revealed, at least a portion of the header itself can be encrypted with a cipher. Depending on an exact implementation, the cipher for the header may or may not be identical to the one used for the document. The key (referred to as a user key) to decrypt the encrypted header can, for example, be stored in a local store of a terminal device and activated only when the user associated with it is authenticated. As a result, only an authorized user can access the secured document.

[0040] Optionally, the two portions (i.e., the header (possibly encrypted) and the encrypted document) can be encrypted again and only decrypted by a user key. In another option, the encrypted portions (either one or all) can be error-checked by an error-checking portion, such as using a cyclical redundancy check to ensure that no errors have been incurred to the encrypted portion(s) or the secured document.

[0041] FIG. 2B is a block diagram of a secured item access system 240 according to one embodiment of the invention. The secured item access system 240 operates to process a secured item 242 on behalf of a requestor to either permit or deny access to its contents. The secured item 242 is, for example, a secured file, such as a secured document.

[0042] The secured item 242 includes a header portion 244 and an encrypted data portion 246. The header portion 244 includes at least a pointer 248. When a requestor requests access to the secured item 242, the pointer 248 from the header portion 244 is supplied to a storage device 250. The pointer 248 is used to locate security information 252 stored in the storage device 250. In this embodiment, the security information 252 is not encrypted; however, in another embodiment, the security information 252 could be further secured by encryption. Hence, the pointer 248 is used to retrieve the security information 252 from the storage device 250.

[0043] The header portion 244 also includes at least an encrypted file key 254. The encrypted file key 254 is encrypted in this embodiment to secure the file key. Hence, the encrypted file key 254 is supplied to a first decryption module 256. The first decryption module 256 also receives a user key. In one embodiment, the user key is a private key, and in another embodiment, the user key is a public key. In any case, the first decryption module 256 operates to decrypt the encrypted file key 254 using the user key and thus produces an unencrypted file key 258.

[0044] The security information 252 typically includes at least access rules for access to the encrypted data portion 246 of the secured item 242. The security information 252 is supplied to an access rules analyzer 260. The access rules analyzer 260 also receives user privileges associated with the requestor. The access rules analyzer 260 examines the user privileges and the security information 252, namely, the access rules contained therein, to determine whether the requestor has sufficient privileges to gain access to the encrypted data portion 246 of the secured item 242. The access rules analyzer 260 outputs an access decision to an access controller 262. The access controller 262 receives the access decision and the file key 258. When the access controller 262 determines that the access decision does not permit the requestor to gain access to the encrypted data portion 246 of the secured item 242, then access to the encrypted data portion 246 for the secured item 242 is denied. Alternatively, when the access controller

262 determines that the access decision does permit the requestor to gain access to the encrypted data portion 246 of the secured item 242, then the file key 258 is supplied to a second decryption module 262. In addition, the encrypted data portion 246 of the secured item 242 (i.e., the data of the secured item 242) is supplied to the second decryption module 264. The second decryption module 264 then operates to decrypt the encrypted data portion 246 using the file key 258 to produce an unencrypted data portion 266. The unencrypted data portion 266 is then made available to the requestor, thereby permitting the requestor to gain access to the data associated with the secured item 242.

[0045] According to the invention, the header portion of a secured item is able to be reduced in size due to the use of a pointer. More particularly, the pointer in the header portion points to separately stored security information. Since the size of the pointer is substantially smaller than the size of the security information pointed to, the overall size of the secured item is reduced. In one embodiment, the pointer is structured in a fixed number of bits so that the size of the pointer is constant.

[0046] Additionally, with security information being separately stored, the security information is able to be shared across different documents, thus reducing the storage burdens for storage of secured items. Changes or modifications to security rules or other security information can be more easily made because changes to the secured items themselves are not necessary. That is, changes to security information stored to a storage device are performed without alterations to the corresponding secured items.

[0047] With respect to manageability of secured items, one feature of the invention is that through use of pointers to security information (stored at a storage device separately from the secured files) different secured files are able to share the same stored security information or parts thereof. For example, multiple secured files can utilize identical pointers such that they all share the same security information stored on a local storage device. Consequently, managing the security provided to the secured files is at least in part dependent upon the security information. Hence, by being able to share common security information, not only can the amount of security information storage space being utilized be substantially reduced, but also access to the associated secured files can be managed more efficiently.

[0048] FIG. 2C is a diagram of a representative data structure 280 for a secured file. For example, the secured file can be the secured item 242 illustrated in FIG. 2B. The data structure 280 includes a header (or header portion) 282 and an encrypted data portion 284. The header 282 includes a flag bit 286, at least one pointer 288, and an encrypted file key 290. The flag bit 286 indicates whether or not the data structure pertains to a file that is secured. The at least one pointer 288 points to a remote data structure 292 stored in a storage device. The storage device is typically a local storage device. In other words, the data structure 280 and the remote data structure 292 are typically stored on a common machine (e.g., desktop or portable computer). The data structure 292 stores security information 294. The data structure 292 storing the security information 294 can vary depending upon implementation. However, as shown in FIG. 2C, the data structure 292 for the security information 294 includes a user identifier (ID) 296-1, rules (access rules) 296-2 and other 296-3. The other 296-3 is additional space for other information to be stored within the security information 294. For example, the other information 296-3 may be used to include other information facilitating secure access to the secured file, such as version number or author identifier. The encrypted file key 290 is normally itself decrypted and then used to decrypt the encrypted data portion 214 so as to access the content or data of the secured file.

[0049] FIG. 3 is a flow diagram of secured document access processing 300 according to one embodiment of the invention. The secured document access processing 300 is typically performed by a computer. The computer can be a local computer or a remote computer. Further, the computer can also be considered both a local and a remote computer that operate in a client-server fashion.

[0050] The secured document access processing 300 can be invoked when a requestor selects a document to be accessed. The document is a particular type of file (for example, design.doc). Therefore, more generally, the requestor selects a file in a folder or among other files to be accessed. Once the document has been selected, the secured document access processing 300 is invoked. Initially, the selected document to be accessed is obtained 302. Typically, the selected document will have a header portion and a data portion. Next, a security information pointer is retrieved 304 from the header portion of the selected document. Here, the header portion of the selected document includes at least the security information pointer that points to an address location where the corresponding security information is located. Next, the security information for the selected document is obtained 306 using the security information pointer.

[0051] A decision 308 then determines whether the security information permits the requested access. In this regard, the security information may or may not be encrypted so that it remains secure while stored on the local storage. If the security information is encrypted, the security information would be decrypted (e.g., through use of a user key) to gain access to the security information. The security information contains, among other things, access rules. These access rules are used by the decision 308 in determining whether the requested access is permitted. Namely, the access rules are compared to privileges associated with the requestor for the selected document. When the decision 308 determines that the security information (namely the access rules) does not permit the requested access, then an

access denied message is provided 310 to the requestor. On the other hand, when the decision 308 determines that the security information does permit the requested access, then the data portion of the selected document is decrypted 312. Typically, the header portion also includes a file key. The file key is itself normally encrypted and can be decrypted with a user key. The file key can be used to decrypt the encrypted data portion of the selected document. Once the data portion has been decrypted, the data portion is provided 314 to the requestor. Here, the requestor has gained access to the selected document. Following the operation 314, as well as following the operation 310, the secured access document processing 300 is complete and ends.

[0052] The security information is stored to a storage device and located through use of a pointer that is provided with a header portion of a secured file (document). The manner in which the security information is stored within the storage device can vary depending upon implementation. In one embodiment, the pointer directly points to a storage location (i.e., memory location) within the storage device. Stored at the storage location designated by the pointer is the security information. As an example, the security information 294 pointed to by the pointer 288 shown in FIG. 2C can be located in this manner.

[0053] In another embodiment, the security information can be stored in a database-type organization. FIG. 4A illustrates a data organization 400 according to one embodiment of the invention. According to the data organization 400, the association between a secured file 402 and its associated security information as separately stored in a storage device is illustrated. The secured file 402 includes a header portion 404 and a data portion 406. The header portion 404 includes at least a pointer 408 that points to a security information table 410. The security information table 410 can then, in turn, include a pointer to a rules table 412 that can store access rules. In general, the security information table 410 is encrypted by one or more keys in a key table 414. The tables 410, 412 and 414 of the data organization 400 can be provided within a database.

[0054] FIG. 4B illustrates exemplary tables for the security information table 410, the rules table 412, and the key table 414 illustrated in FIG. 4A. The security information table 410 is a main table that is addressed through use of a memory address maintained by an operating system. The pointer 408 points to one of the rows, namely, memory addresses (e.g., operating system addresses), of the security information table 410. The rows of the security information table 410 in turn can point to other tables or include data therein. As shown in FIG. 4B, the columns of the security information table 410 include an operating system address, a policy identifier column and an owning group identifier column. The operating system address column can serve as an index to the security information table 410. The policy identifier column of the security information table 410 includes pointers to particular rows in the rule table 412. The rules within the rules table 412 can be provided within a variety of different formats. The rules table 412 shown in FIG. 4B provides the rules expressed in a markup language format (such as eXtensible Markup Language (XML)). The owning group identifier column of the security information table 410 includes pointers pertaining to group identifiers. These pointers pertaining to the group identifiers point to rows within the key table 414. As shown in FIG. 4B, each row within the key table 414 can store a public key and a private key. In other words, each of the group identifiers is associated with a pair of public and private keys. Depending on implementation and a specific stage of a secured file being accessed, one or both of the public and private keys may be encrypted or readily used to encrypt or decrypt the security information table 410. In any case, only an authenticated user (in a user group identified by one of the group identifiers) can retrieve one of the keys to access the secured file 402.

[0055] FIG. 5 is a block diagram of a file security management system 500 according to one embodiment of the invention. The file security management system 500 includes a security information manager 502 and a database 504. An administrator 506 interacts with the security information manager 502 typically through a graphical user interface (GUI). In this manner, the administrator 506 is able to store, modify or delete information in the database 504. By altering the security information stored in the database 504, the administrator is able to manage the nature of the security provided to associated files or documents. The database 504 includes the security information arranged in a plurality of tables. The tables include a security information table 508, a rules table 510 and a key table 512. For example, using the security information manager 502, the administrator 506 can provide new keys for the key table 512, such as to rotate keys for security reasons. As another example, the administrator 506 might interact with the security information manager to store new access rules (or policies) to the rules table 510. Hence, the separate and distributed storage of the security information and the user of pointers provides an efficient data arrangement that allows security information to be efficiently stored, modified and shared.

[0056] In general, a secured item (e.g., secured file) with its pointer to separately stored security information are stored relative to one another. Hence, moving the location of the secured item requires adjustment to the pointers, particularly when the movement is to another storage device. In other words, although stored in an efficient and manageable format, the secured item and its security information are not readily portable relative to one another as an association to the security information must be maintained. Processing discussed below is able to provide temporary portability for the movement of the secured item.

[0057] FIGs. 6A and 6B are flow diagrams of secured

file portability processing 600 according to one embodiment of the invention. The secured file portability processing 600 pertains to processing carried out on a secured file when such file is being moved from its present storage device to a different storage device. Typically, these different storage devices are associated with different computers. The processing shown in FIG. 6A is typically performed by the computer initially storing the secured file, and the processing shown in FIG. 6B is typically performed by a different computer.

[0058] The secured file portability processing 600 begins with a decision 602 that determines whether a request to move a secured file to a different storage device has been received. When the decision 602 determines that such a request has not yet been received, the secured file portability processing 600 awaits such a request. Once the decision 602 determines that a request to move a secured file to a different storage device has been received, the secured file portability processing 600 begins its processing. Initially, a security information pointer is retrieved 604 from a header portion of the secured file. Security information for the secured file is then obtained 606 using the security information pointer. Typically, the security information is not encrypted at this point. Hence, the security information is encrypted. The security information can be encrypted using a key (e.g., public key). Next, the security information pointer in the header is replaced 608 with the encrypted security information.

[0059] After the security information pointer has been replaced with the encrypted security information, the secured file is portable and can thus be moved 610 to the different storage device. Once moved, the secured file can be stored to the different storage device in its portable format or additional processing as provided in FIG. 6B can be performed to store the secured file in a more efficient and manageable format.

[0060] When such additional processing is performed, the encrypted security information is initially retrieved 612 from the header of the secured file. The encrypted security information is then decrypted 613. Once decrypted 613, the security information is stored 614 to the different storage device. The different machine that is performing the processing shown in FIG. 6B has the security information stored therein. Next, a pointer to the stored location of the security information is generated 616. Thereafter, the encrypted security information in the header is replaced 618 with the pointer. At this point, the secured file has been altered such that it includes the pointer and not the encrypted security information. Consequently, the secured file is now more efficiently stored and more manageable. Following the operation 618, the secured file portability processing 600 is complete and ends.

[0061] The invention is preferably implemented by software, but can also be implemented in hardware or a combination of hardware and software. The invention can also be embodied as computer readable code on a computer readable medium. The computer readable medium is any data storage device that can store data which can thereafter be read by a computer system. Examples of the computer readable medium include read-only memory, random-access memory, CD-ROMs, DVDs, magnetic tape, optical data storage devices, and carrier waves. The computer readable medium can also be distributed over network-coupled computer systems so that the computer readable code is stored and executed in a distributed fashion.

[0062] The various embodiments, implementations and features of the invention noted above can be combined in various ways or used separately. Those skilled in the art will understand from the description that the invention can be equally applied to or used in other various different settings with respect to various combinations, embodiments, implementations or features provided in the description herein.

[0063] The advantages of the invention are numerous. Different embodiments or implementations may yield one or more of the following advantages. One advantage of the invention is that access rules or criteria are able to be stored separate from the corresponding secured items. Another advantage of the invention is that security information to be used with secured items is able to be readily altered by a security administrator. Still another advantage of the invention is that centralized, dynamic security management is facilitated. Yet another advantage of the invention is that the security approaches of the invention are useful for not only files but also non-file resources, even non-encryptable resources such as pipes/streams, ports and devices.

[0064] The foregoing description of embodiments is illustrative of various aspects/embodiments of the present invention. Various modifications to the present invention can be made to the preferred embodiments by those skilled in the art without departing from the true spirit and scope of the invention as defined by the appended claims. Accordingly, the scope of the present invention is defined by the appended claims rather than the foregoing description of embodiments.

Claims

1. A method for accessing a secured file, said method comprising:-

   (a) obtaining the secured file to be accessed, the secured file having a header portion and a data portion;
   (b) retrieving a security information pointer from the header portion of the secured file;
   (c) obtaining for the secured file security information pointed to by the security information pointer; and
   (d) permitting access to the secured file to an extent permitted by the security information.

2. A method as recited in claim 1, wherein said permitting step (d) comprises:-

(d1) retrieving a file key from the header portion; and
(d2) decrypting the data portion of the secured file using the file key.

3. A method as recited in claim 1 or 2, wherein a requestor desires to access the secured file, the requestor having requestor characteristics;
wherein said permitting step (d) comprises:-

retrieving at least one access rule from the security information; and
determining whether the requestor is permitted to access the secured file based on the at least one access rule and the requestor characteristics.

4. A method as recited in any preceding claim, wherein a requestor desires to access the secured file, and wherein said method further comprises:-

(e) decrypting, following said obtaining step (c) and before said permitting step (d), the security information.

5. A method as recited in claim 4, wherein said decrypting step (e) of the security information is performed using a key associated with the requestor and the key is a user key.

6. A method as recited in claim 4 or 5, wherein the security information is located locally in a client machine from which the secured file is being accessed or remotely in a computing machine coupled to the client machine over a network.

7. A computer readable medium including at least computer program code for accessing a secured item, said computer readable medium comprises:-

computer program code for obtaining the secured item to be accessed, the secured item having a header portion and a data portion;
computer program code for retrieving a security information pointer from the header portion of the secured item;
computer program code for obtaining for the secured item security information pointed to by the security information pointer; and
computer program code for permitting access to the secured item to an extent permitted by the security information.

8. A computer readable medium as recited in claim 7, wherein said computer program code operates the method of any one of claims 2 to 6.

9. A system for accessing a secured item, the secured item having a header portion and an encrypted data portion, the header portion including a pointer and an encrypted key, said system comprising:-

a storage device, said storage device storing security information for a plurality of different secured items, the pointer serving to locate the security information associated with secured item;
a first decryption module, said first decryption module receiving the encrypted key from the header portion of the secured item and decrypting the encrypted key to obtain a key;
an access analyzer operatively connected to said storage device, said access rules analyzer determines whether the encrypted data portion is permitted to be accessed by a requestor based on the security information; and
a second decryption module operatively connected to said access analyzer, said second decryption module decrypting the encrypted data portion using the key to produce an unencrypted data portion that the requestor is able to access, provided said access analyzer determines that the encrypted data portion is permitted to be accessed by the requestor.

10. A system as recited in claim 9, wherein the security information includes at least an access rule; wherein the requestor has user privileges associated therewith; and wherein said access analyzer determines whether the encrypted data portion is permitted to be accessed by the requestor based on the access rule and the user privileges.

11. A data structure for a secured file, said data structure comprising:-

a header portion containing at least a pointer to separately stored security information and a key, at least the key portion of said header portion is encrypted; and
a data portion containing at least encrypted data of the secured file.

12. A data structure as recited in claim 11, wherein the pointer is used to access the separately stored security information which is in turn used to determine whether a particular requestor for access to the secured file is permitted, and then when access is permitted, providing the key to the requestor so that the encrypted data in said data portion can thereafter be decrypted and thus accessed.

13. A data structure as recited in claim 11 or 12, wherein

the pointer points to a database that stores the separately stored security information for a plurality of secured files.

14. A data structure as recited in claim 12 or 13, wherein like separately stored secured information can be shared by a plurality of secured files.

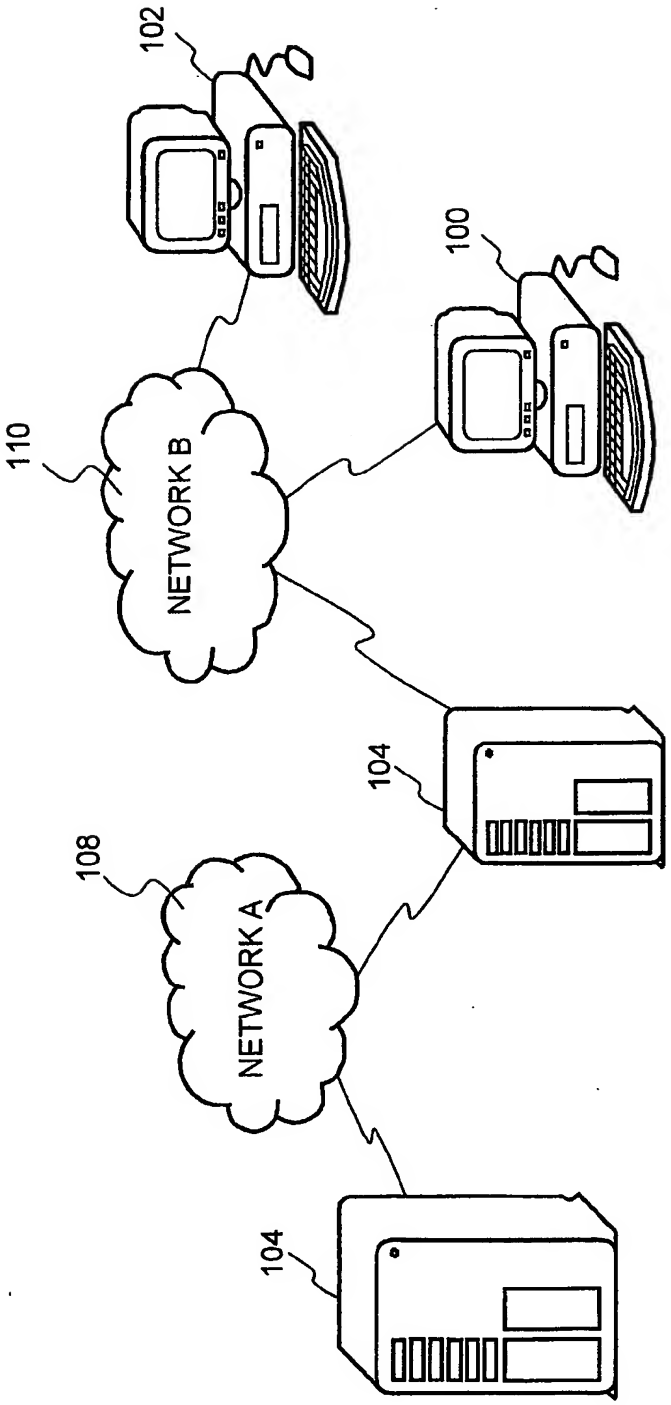

FIG. 1A

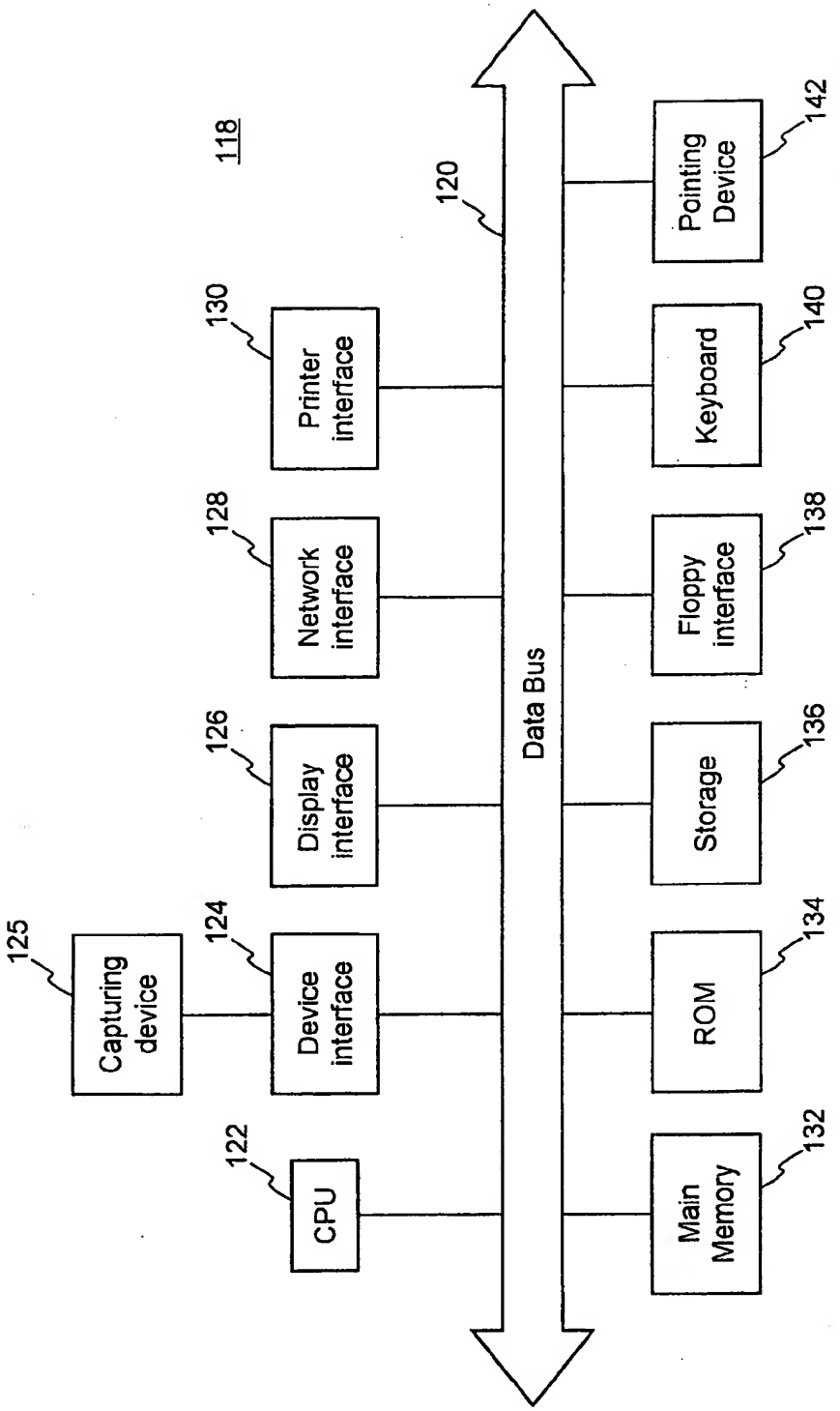

FIG. 1B

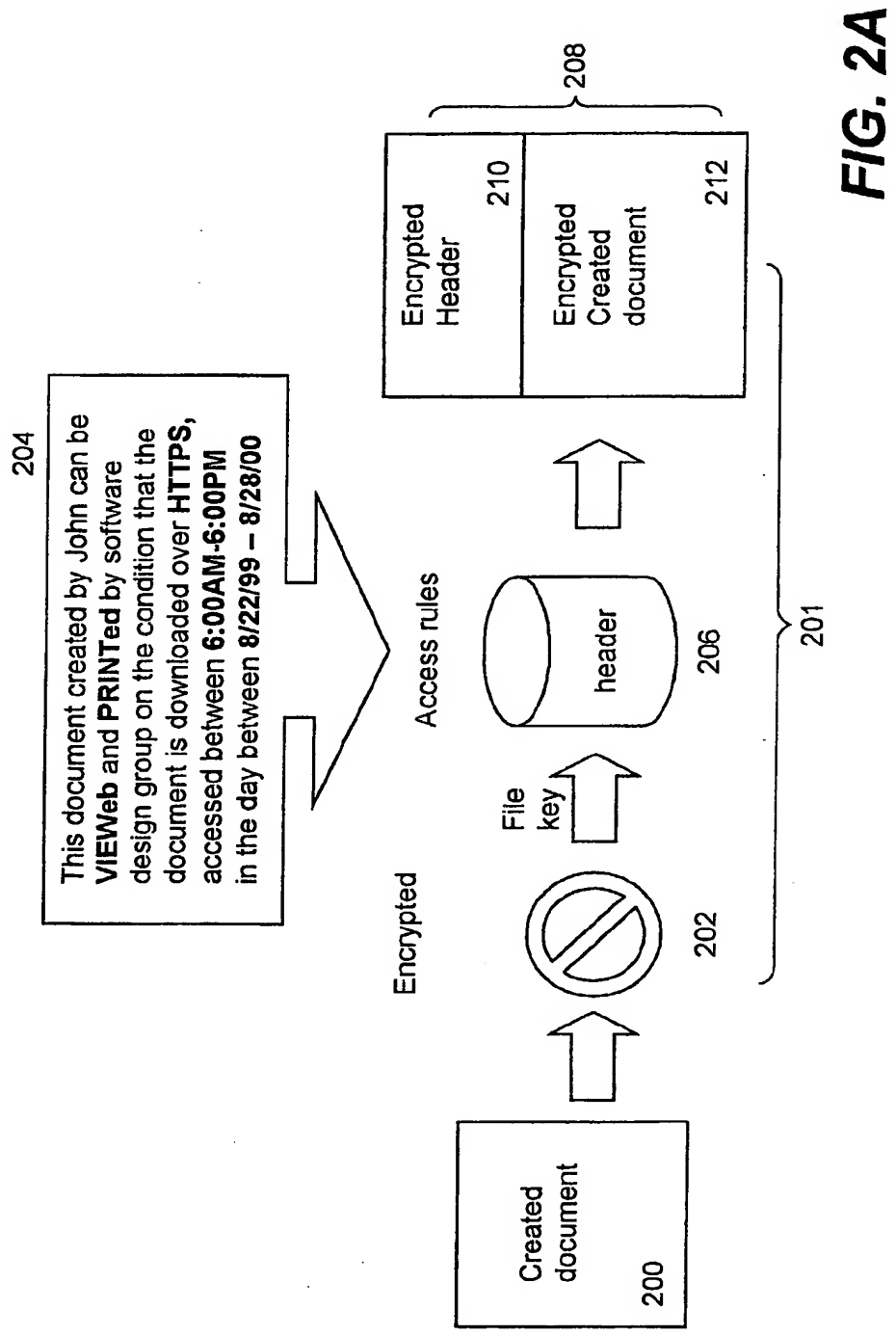

FIG. 2A

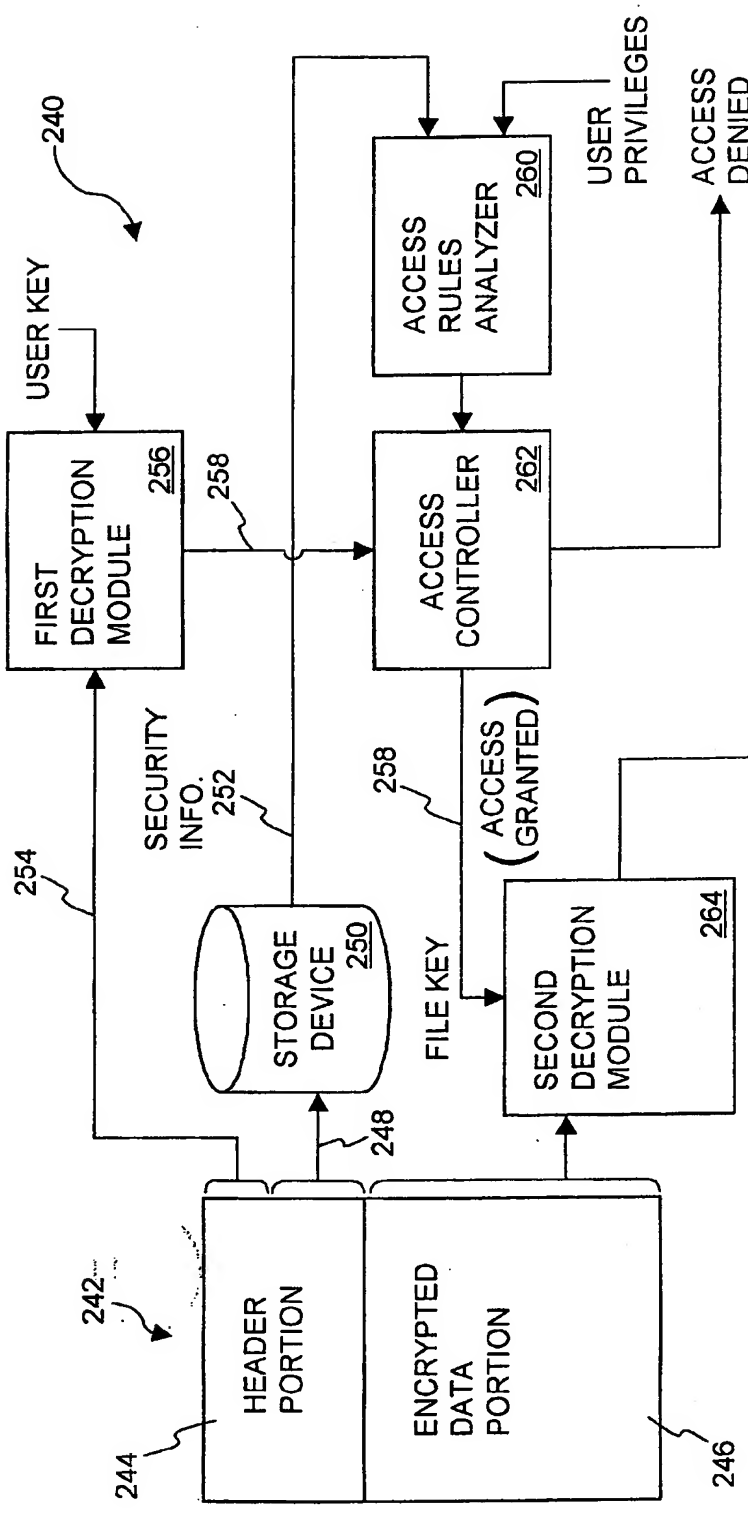

FIG. 2B

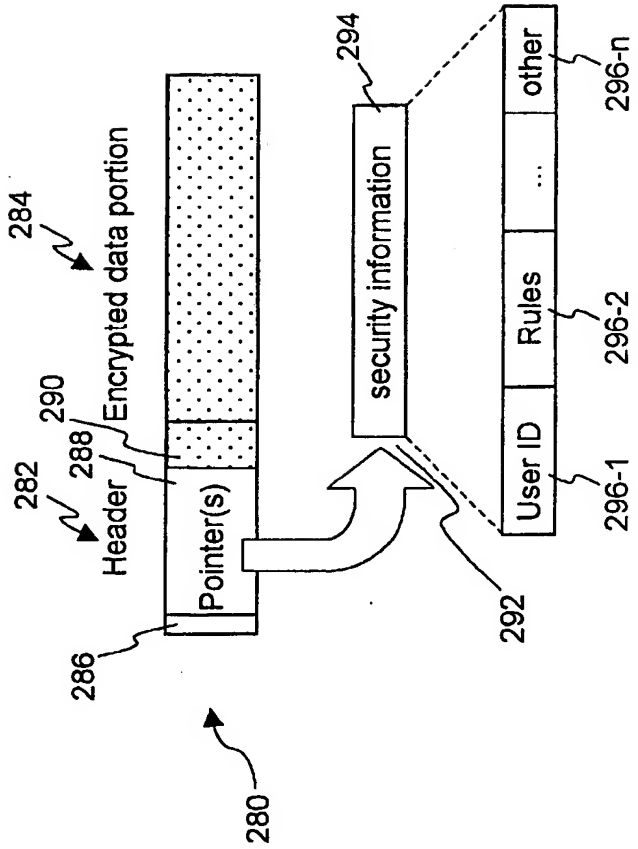

FIG. 2C

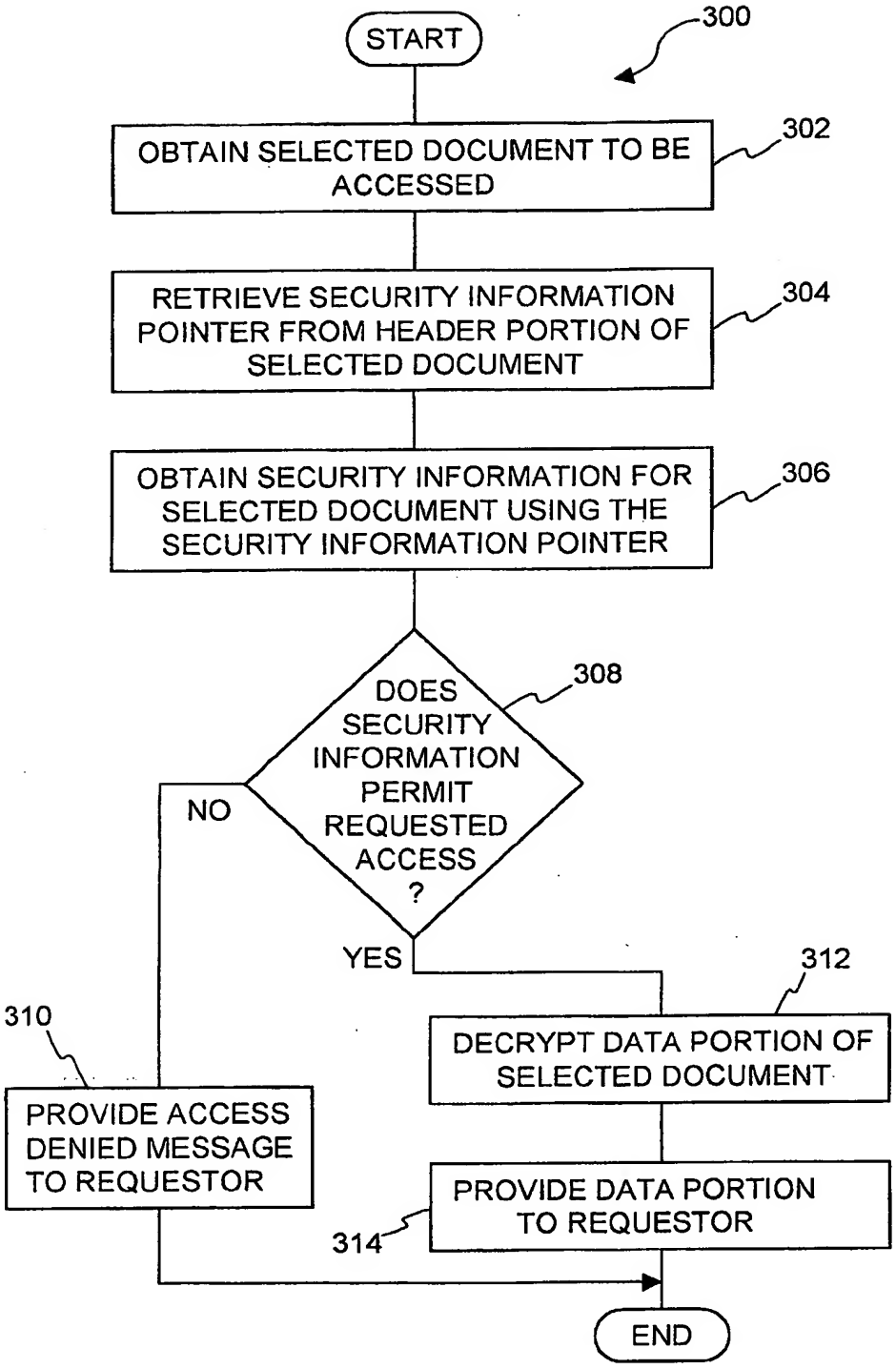

FIG. 3

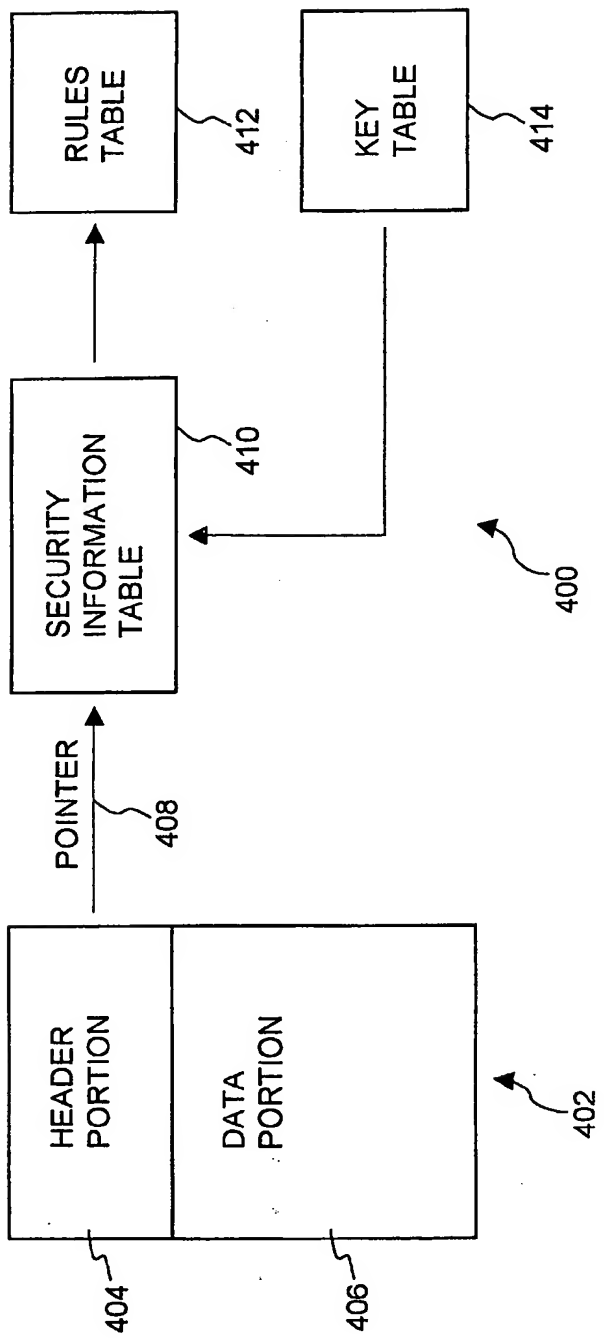

FIG. 4A

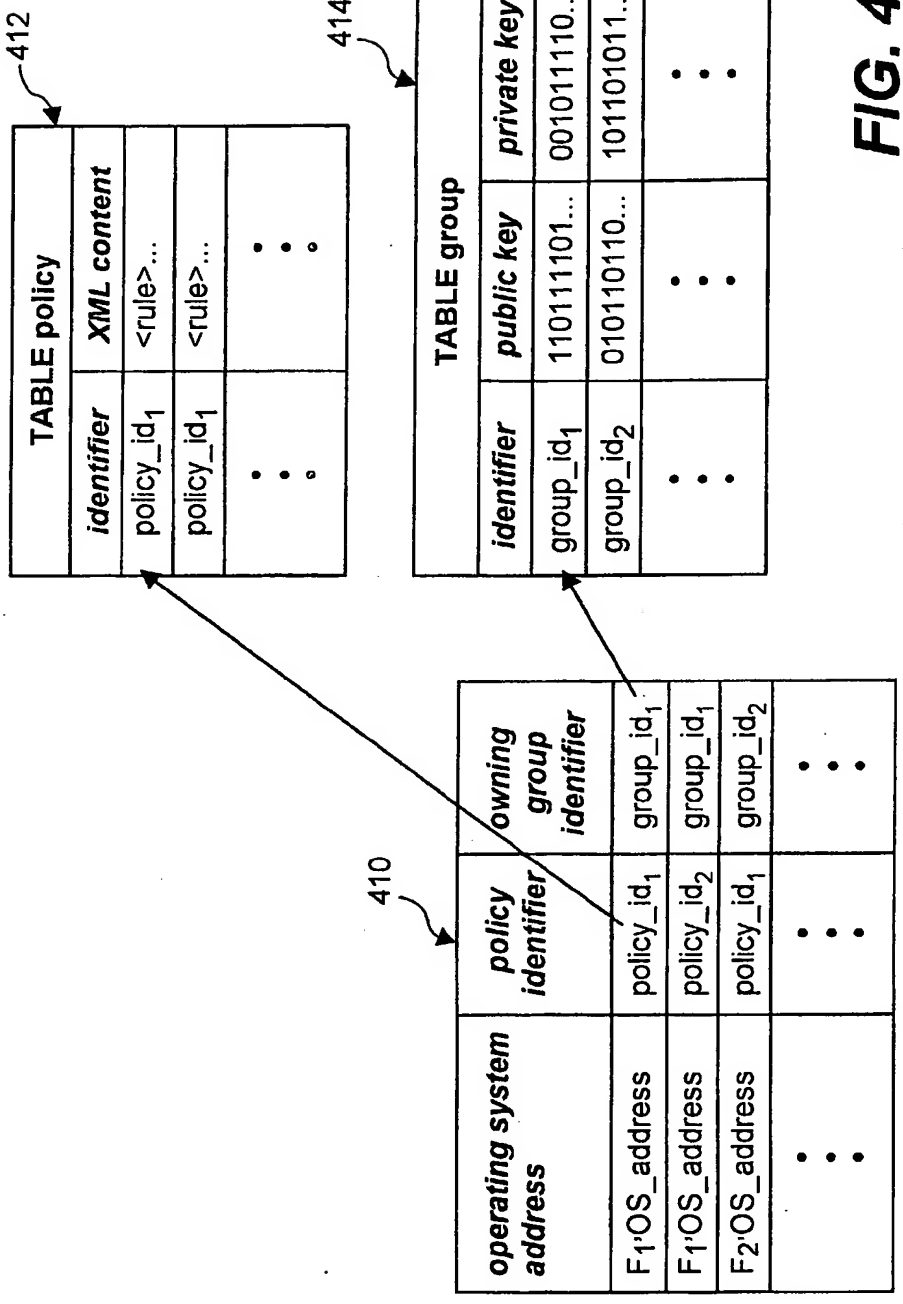

FIG. 4B

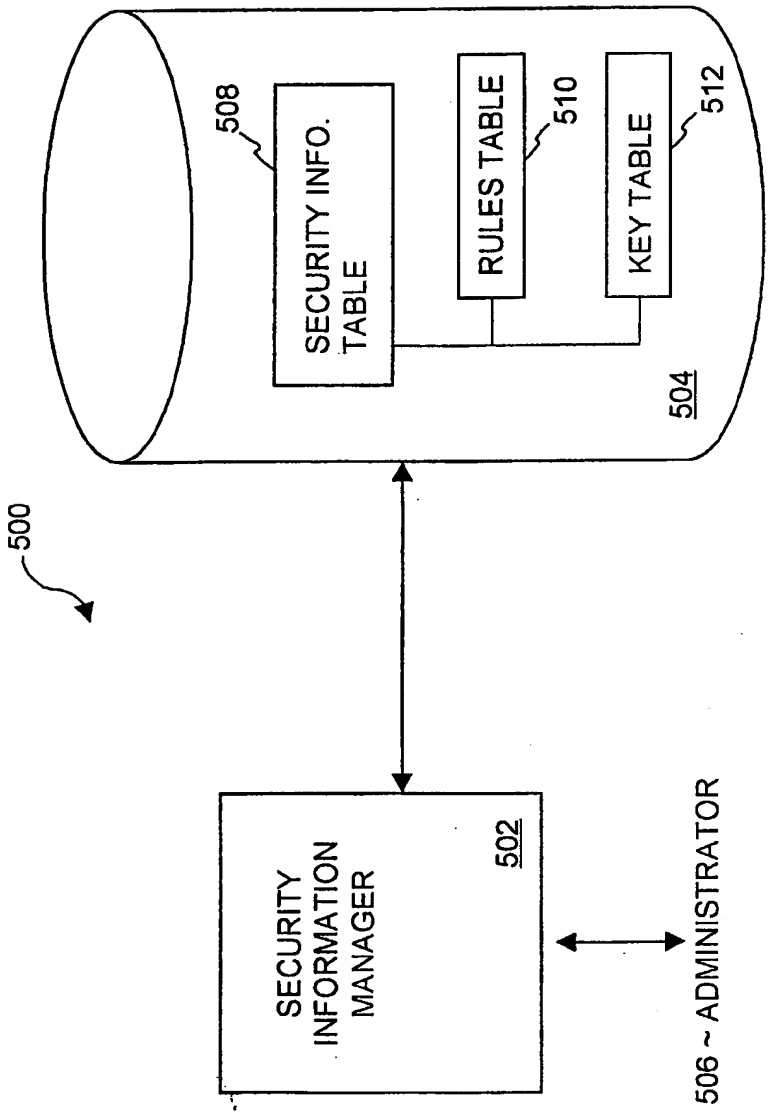

FIG. 5

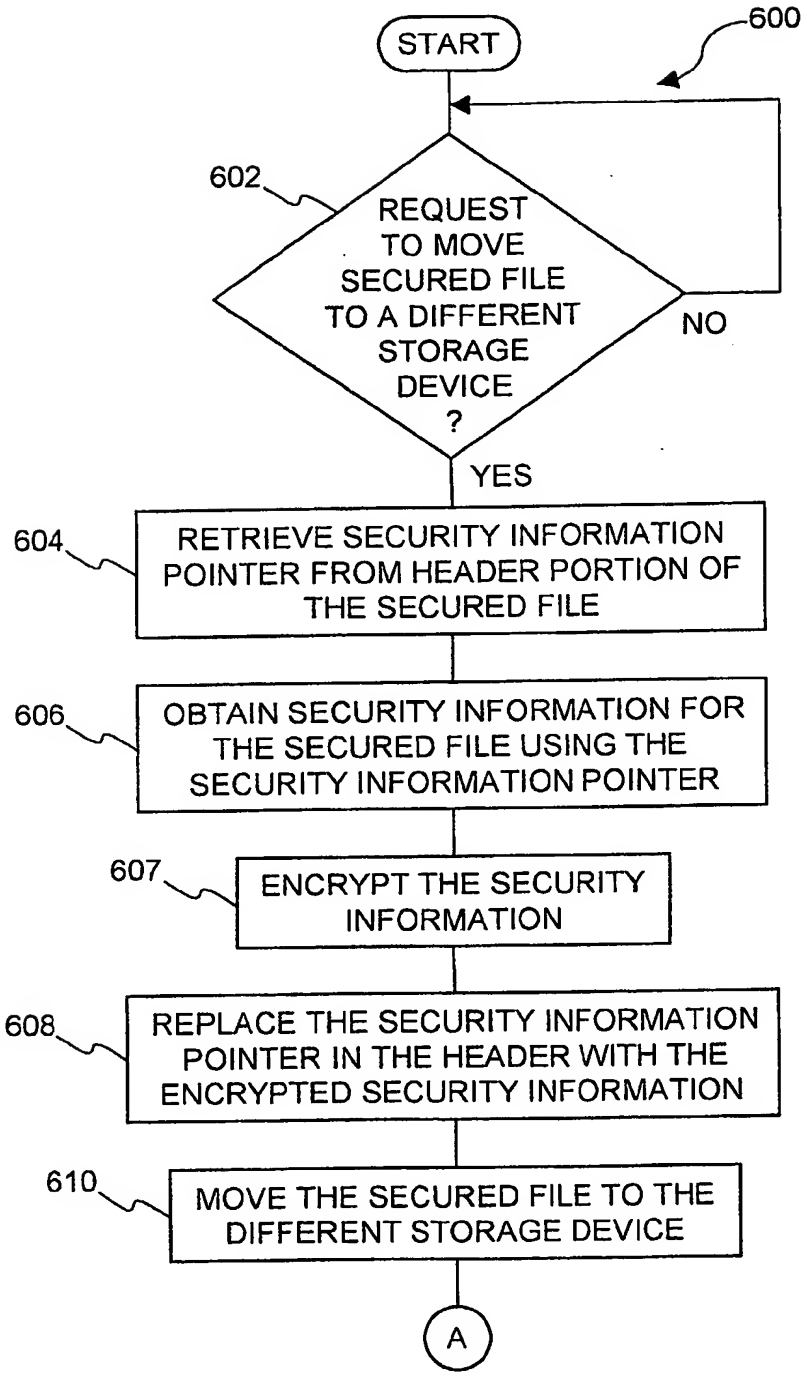

FIG. 6A

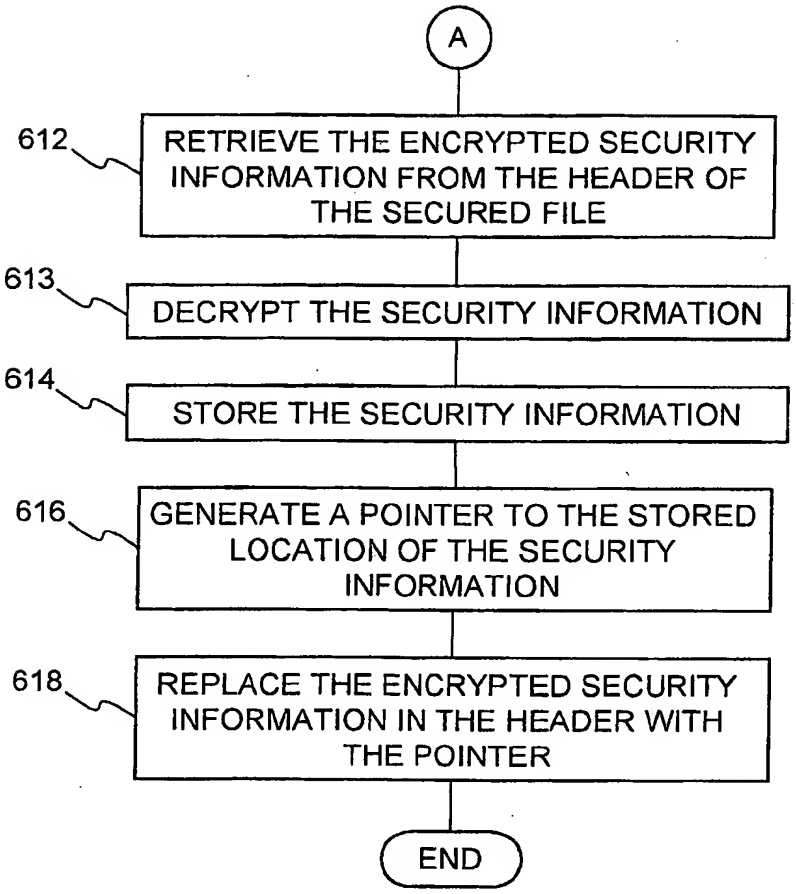

FIG. 6B